In the Specification

Please replace paragraph [0032] with the following new paragraph:

[0032] Figure 4 shows a RAT 418 that includes three component RATs: a high-bandwidth RAT 422, a mid-bandwidth RAT 424, and a low-bandwidth RAT 426. As shown in Figure 4, trace cache 410 is coupled [[t]] to RAT 418. Re-scheduler 440 is coupled to RAT 418, as shown in Figure 4. The high-bandwidth RAT 422 and low-bandwidth RAT 426 are shown with the corresponding number of read ports 428, 436, respectively, and write ports 430, 438, respectively, as used by the high-bandwidth RAT 222 and low-bandwidth RAT 224 of Figure 2. However, in other embodiments other numbers of read ports and write ports may be used. The mid-bandwidth RAT 424 may have a number of read ports 432 and a number of write ports 434 somewhere between that used by the high-bandwidth RAT 422 and the low-bandwidth RAT 426. In the Figure 4 embodiment, mid-bandwidth RAT 424 is shown with N read ports 432 and N/2 write ports 432, although other numbers could be chosen.

In the Claims

Please amend the claims as follows.

For the Examiner's convenience, a list of all claims is included below.

1. (Currently Amended) A processor, comprising:

a first register alias table including a first number of read ports to translate a first set of logical register addresses to physical register addresses; a second register alias table including a

second number of read ports to translate a second set of logical register addresses to physical register addresses, wherein said first number that includes at least one read port for each source operand for an instruction is greater than said second number; and

a pipeline logic to stall a pipeline when a first instruction utilizes more logical register addresses from said second set than said second number.

2. (Original) The processor of claim 1, wherein said first number is proportional to a third number of logical register addresses in said first set.

3. (Canceled).

4. (Original) The processor of claim 1, further comprising a trace cache to supply a trace of micro-operations to said first register alias table and said second register alias table.

5. (Original) The processor of claim 4, wherein said trace cache includes trace cache logic to build said trace limiting a third number of live-in and live-out logical registers to said second number.

6. (Currently Amended) A method, comprising:

storing frequently used translations from logical register addresses to physical register addresses in a first register alias table; storing less-frequently used translations from logical register addresses to physical register addresses in a second register alias table, where said second register alias table has fewer read ports than said first register alias table that includes at least one read port for each source operand for an instruction; and

stalling a pipeline when a first number of logical register addresses is supplied to said second register alias table, and said first number is greater than a second number of read ports of said second register alias table.

7. (Original) The method of claim 6, wherein said storing less-frequently used translations includes identifying said less-frequently used translations from a set of logical register addresses.

8. (Original) The method of claim 7, wherein said identifying includes selecting infrequently used temporary registers.

9. (Original) The method of claim 8, wherein said infrequently used temporary registers are associated with a long micro-operation flow.

10. (Original) The method of claim 7, wherein said identifying includes selecting control registers.

11. (Original) The method of claim 10, wherein said identifying includes choosing registers used by a compiler.

12. (Original) The method of claim 6, further comprising building a trace in a trace cache whose micro-operations require no more live-in registers and live-out registers using said second register alias table than a first number of read ports of said second register alias table.

13. (Original) The method of claim 12, wherein said building includes permitting no more live-out registers using said second register alias table than a second number of write ports of said second register alias table.

14. (Canceled)

15. (Currently Amended) An apparatus, comprising:

means for storing frequently used translations from logical register addresses to physical register addresses in a first register alias table; and

means for storing less-frequently used translations from logical register addresses to physical register addresses in a second register alias table, where said second register alias table has fewer read ports than said first register alias table that includes at least one read port for each source operand for an instruction; and

means for stalling a pipeline when a first number of logical register addresses is supplied to said second register alias table, and said first number is greater than a second number of read ports of said second register alias table.

16. (Original) The apparatus of claim 15, wherein said means for storing less-frequently used translations includes means for identifying said less-frequently used translations from a set of logical register addresses.

17. (Original) The apparatus of claim 15, further comprising means for building a trace in a trace cache whose micro-operations require no more live-in registers and live-out registers using said second register alias table than a first number of read ports of said second register alias table.

18. (Original) The apparatus of claim 17, wherein said means for building includes means for permitting no more live-out registers using said second register alias table than a second number of write ports of said second register alias table.

19. (Canceled).

20. (Currently Amended) A system, comprising:

a processor including a first register alias table including a first number of read ports to translate a first set of logical register addresses to physical register addresses, and a second register alias table including a second number of read ports to translate a second set of logical register addresses to physical register addresses, wherein said first number that includes at least one read port for each source operand for an instruction is greater than said second number;

an audio input/output device; an interface to couple said processor to said audio input/output device; and

a pipeline logic to stall a pipeline when a first instruction utilizes more logical register addresses from said second set than said second number.

21. (Original) The system of claim 20, wherein said first number is proportional to a third number of logical register addresses in said first set.

22. (Canceled)

23. (Original) The system of claim 20, further comprising a trace cache to supply a trace of micro-operations to said first register alias table and said second register alias table.

24. (Original) The system of claim 23, wherein said trace cache includes trace cache logic to build said trace limiting a third number of live-in and live-out logical registers to said second number.